Totality in arena games

Pierre Clairambault, Russ Harmer

▶ **To cite this version:**

Pierre Clairambault, Russ Harmer. Totality in arena games. Annals of Pure and Applied Logic, 2009, 161 (5), pp.673-689. 10.1016/j.apal.2009.07.016 . hal-00443535

HAL Id: hal-00443535
https://hal.science/hal-00443535

Submitted on 30 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Totality in arena games

Pierre Clairambault, Russ Harmer*

PPS, CNRS & Université Paris Diderot–Paris 7, Case 7014, 75205 Paris Cedex 13

Abstract

We tackle the problem of preservation of totality by composition in arena games. We first explain how this problem reduces to a finiteness theorem on what we call *pointer structures*, similar to the parity pointer functions of Harmer, Hyland & Melliès and the interaction sequences of Coquand. We discuss how this theorem relates to normalization of linear head reduction in simply-typed λ-calculus, leading us to a semantic realizability proof *à la* Kleene of our theorem. We then present another proof of a more combinatorial nature. Finally, we discuss the exact class of strategies to which our theorems apply.

1. Introduction

Over the last fifteen years, game semantics has been extensively used to give accurate models of a wide variety of primarily sequential programming language features; see [2, 3, 7, 20, 22, 24] among others, and logics [5, 10, 16, 18, 23]. In this type of model, types/formulae are represented by *games* for two protagonists, Opponent and Player, and programs/proofs by *strategies* (for Player). In almost all cases, these models are fully abstract/complete, meaning that all strategies of the model correspond to something in the language/logic: there is no "junk". This approach has enabled a semantic classification of programming languages according to the particular combinations of constraints they require on strategies in order to achieve full abstraction.

One particularly important constraint, called *innocence*, corresponds to the property of a programming language being applicative/functional. Clearly not all programming languages satisfy innocence but, in contrast, all logics studied to date *do* have this property. Another constraint on strategies, of a rather different nature, is *totality*—where the strategy always has a response, no matter what Opponent does. A general programming language never satisfies this but mathematical logics always do. A game semantics of a logic should thus consist of a category of games and total innocent strategies where the fact that total strategies compose corresponds to normalization/cut elimination of the logic.

*Corresponding author

Email addresses: pierre.clairambault@pps.jussieu.fr (Pierre Clairambault), russ.harmer@pps.jussieu.fr (Russ Harmer)

In principle then, a logic could be shown to be normalizing by proving that it has a sound interpretation in such a category of total strategies. However, in practice, this reasoning is often inverted in that one starts with a logic already known to be normalizing, interprets it with total strategies (not yet known to form a category) and uses soundness and completeness to establish that the total strategies are in fact indeed closed under composition (and so form a category)! This is done because it can be subtle to establish that a desired class of total strategies is actually closed under composition, due to the problem of "infinite chattering" first noticed in CCS [25] and CSP [17]: when two deterministic processes with a private channel enter a "non-responsive" mode where they will thereafter only communicate via that private channel, completely ignoring their other public channels. We find an analogous situation in game semantics since composition of strategies is defined by precisely such a "communication by private channel" mechanism (often referred to as "parallel composition plus hiding"). Thus, even if the composed strategies are both total, they may engage in infinite chattering whereupon the composite strategy will no longer be total.

To date, game models that directly present the category of total strategies typically use an *extrinsic* notion of "winning" [1, 19] to cut down the space of total strategies to one which is closed under composition. (The alternative approach of imposing the so-called "copycat condition" also solves this problem by restricting to an even smaller class of total strategies where one cannot even conceive of infinite chattering and so all strategies are automatically winning, irrespective of what 'winning' means.) It is important to note that the winning condition is independent of innocence; one typically starts with a big category of games and strategies and then identifies a subcategory of total strategies and simultaneously verifies that innocence is preserved. While satisfactory from the purely technical point of view of being able to define the model independently of the logic, one nonetheless feels that this approach could be improved by an *intrinsic* analysis of what constitutes the "good" class of total strategies. In this paper, we provide such an analysis, in particular showing that the class of total, finite innocent strategies forms a category with no need for *any* extrinsic winning condition.

It should be noted that, whereas the original motivation for this work came from game semantics in the Hyland-Ong/Nickau setting [20, 26], the main object of study, *i.e. pointer structures*, is a rather general notion that arises naturally in various forms of games for logic [5] or programming languages [6, 20], as the exponential modalities of linear logic [15] and also in the execution of abstract machines [8]. Consequently, this paper aims to be largely self-contained: no particular knowledge of game semantics should be required to understand either the definition of pointer sequences or the related statements or proofs. Still, some parts of the paper focus on the relation with game semantics, and may require some background. During the development, these parts will be clearly identified. All the necessary background and terminology can be found in [14, 13]. However, the reader unfamiliar with game semantics but interested in pointer sequences nonetheless should be able to skip these parts in a first reading without getting lost.

In Section 2, we start from the "infinite chattering" phenomenon in game semantics and gradually isolate the root cause of this phenomenon, which occurs in pointer structures. We then state our main results. In Section 3, we first sketch a rather circuitous proof that exploits normalization of the λ-calculus. This syntactic proof then informs a semantic proof that uses a realizability argument. By means of contrast, we then present in Section 4 a second proof (of a slightly more general result) of a purely combinatorial nature that was inspired by similar arguments of Coquand [5] and Curien [6]. Our results go beyond simply proving that total, finite innocent strategies compose and, in Section 5, we discuss how our results apply beyond that basic case. Moreover, the two (semantic) proofs provide new, and different, insights into the deeper structure of visible (not just innocent) strategies.

2. Infinite plays and pointer structures

In this section, we reduce the non-compositionality of total strategies in HO/N game semantics to the existence of infinite plays with certain properties. We then gradually strip these infinite plays of all unnecessary information, to get finally to the notion of *pointer structures*. Note that Subsections 2.1 and 2.2 serve to introduce the definitions of pointer structures to the reader already familiar with game semantics. As such, it uses notations and terminology which are not introduced here but can be found in [13]. It should however be possible to jump directly to the formal definition of pointer structures, in Subsection 2.3.

2.1. Infinite chattering

As pointed out in the introduction, total strategies do not compose. A paradigmatic example of this fact is $\delta = \lambda x.xx$. This λ-term is untyped, but one can nonetheless consider its interpretation as an innocent strategy over the universal arena $\mathcal{U}$ [11]. Then one remarks that $[\![\delta]\!]$ is total: it is essentially a combination of copycat strategies. However, as can be witnessed in Figure 1, $[\![(\delta)\delta]\!] := [\![\delta]\!][\![\delta]\!]$ is not total, and even has no response to the initial external Opponent move. This is closely related to the fact that, in the λ-calculus, the class of normalizing terms is not closed under composition. In this section, we investigate the consequences of such a situation on the hidden part of the interaction.

Suppose that $\sigma : A \to B$ and $\tau : B \to C$ are total strategies but that their composite $\sigma;\tau : A \to C$ is not total. This means that there is $s \in \sigma;\tau$ and s' an immediate extension of s such that $\sigma;\tau$ has no response to s'. If we take $u \in \sigma||\tau$ such that $u_{\restriction A,C} = s$ (which exists by the witness property), we notice that the immediate extension s' of s yields an immediate extension u_0 of u. Suppose (WLOG) that the last move added is in A. Since σ is total, it has a response to this last move. This yields an immediate extension u_1 of u_0, which is still in $\sigma||\tau$ but, since s' has no extension in $\sigma;\tau$, this last move of u_1 must be in B. But then τ is total as well and yields a extension u_2 of u_1 which is still in $\sigma||\tau$. By the same argument as above, it too must be in B.

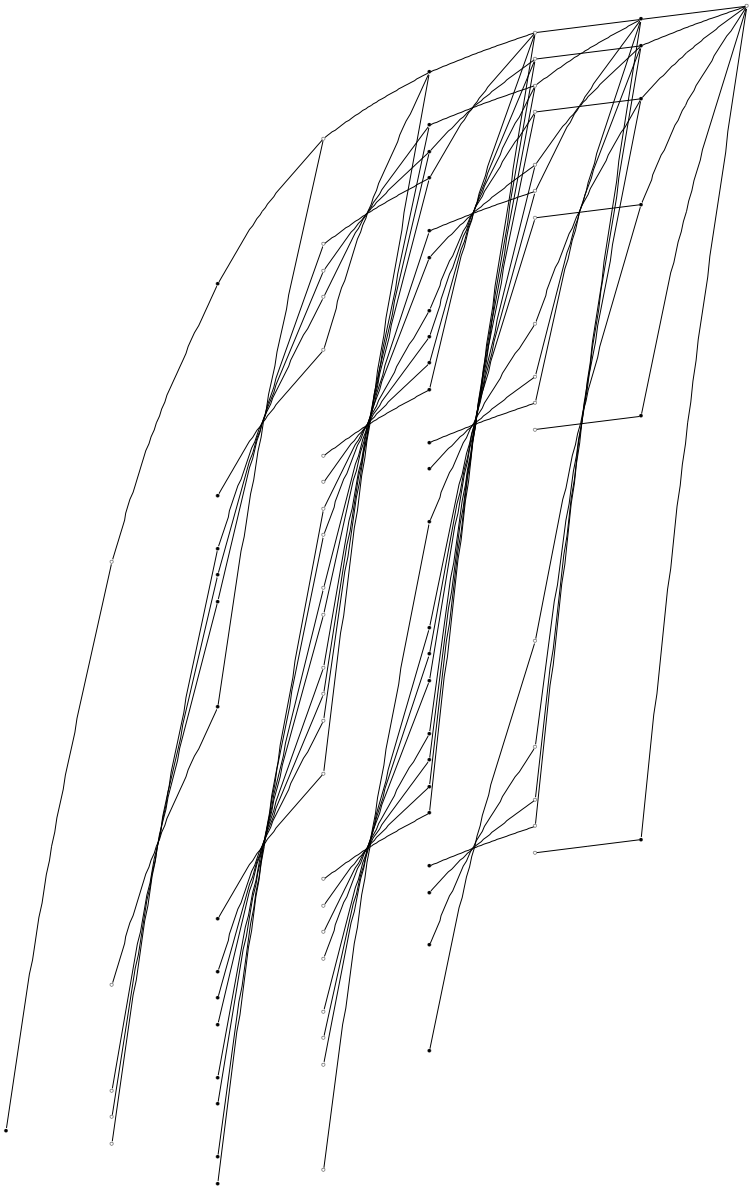


Figure 1: The start of an infinite visible pointer structure with finite memory but infinite depth: the trace of $\delta\delta$.

The reader will easily see what is going on: the process of interaction must build an infinite play in B, accepted by σ and τ, because it can never go back to either A or C but cannot stop either. We have just sketched a proof of the following result:

Proposition 1 (Infinite chattering)**.** *Let $\sigma : A \to B$ and $\tau : B \to C$ be total strategies such that $\sigma;\tau$ is not total. Then there is $u \in \sigma||\tau$ such that $u_{\restriction A,C}$ is finite but $u_{\restriction B}$ is infinite.*

Let us focus on the generated infinite play u. It has the following remarkable property: irrespective of the behaviour of the external Opponent, u eventually becomes both P- and O-visible[1] provided only that σ and τ both satisfy P-visibility. This occurs once the interaction has entered the infinite tail in B whereupon the external Opponent no longer has any influence. A play which is both P-visible and O-visible will be simply called *visible*. A play such as u, which satisfies this property only after a while, will be called *ultimately visible*.

To sum up the above discussion, if the composite $\sigma;\tau : A \to C$ of P-visible total strategies $\sigma : A \to B$ and $\tau : B \to C$ is not itself total, this forces the existence of an infinite play p which is ultimately visible. We now study the conditions under which such a play can exist.

2.2. Collapsing Arenas

In the previous subsection, we showed that to guarantee composition of totality, it suffices to forbid infinite ultimately visible infinite plays. Here, we argue that, in fact, it suffices to ban such plays in *pure* arenas (which only take into account the depth of moves).

For $n \in \mathbb{N}$, let I_n be the following arena:

$$0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow n$$

of length n, and I_ω defined obviously as the projective limit of the sequence $(I_n)_{n\in\mathbb{N}}$, in the category of graphs and graph morphisms[2]. Then plays on I_ω correspond precisely to what we will call pointer structures.

If A is an arena, define a graph morphism $\rho_A : A \to I_\omega$ by sending each node x to the unique node of I_ω of the same depth and embedding all edges from x to y to the unique edge from $\rho_A(x)$ to $\rho_A(y)$. This ρ_A can then be freely extended to ρ_A^*, acting on the legal plays $\mathcal{L}_A$ of A (defined in [13]):

$$\rho_A^* = \left\{ \begin{array}{rcl} \mathcal{L}_A & \to & \mathcal{L}_{I_\omega} \\ \epsilon & \mapsto & \epsilon \\ sa & \mapsto & \rho_A^*(s)\rho(a) \end{array} \right.$$

where $\rho(a)$ points in the same way as a. This is guaranteed to be possible since ρ_A is a graph morphism.

[1] A play is P- (resp. O-)visible iff all its P- (resp. O-)moves point in their P (resp. O-)view.
[2] I_ω is nothing but the infinitary pure arena.

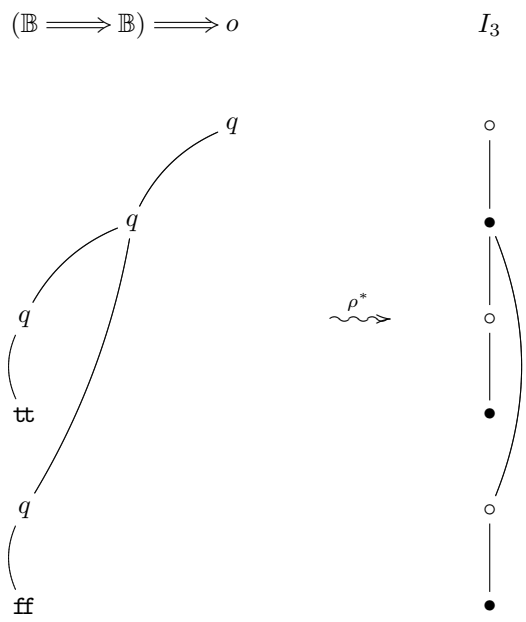


Figure 2: How collapsing creates innocence

The operation of taking ρ_A amounts to *collapsing the arena*: all moves at the same depth are merged. It is immediate to see that all notions on legal plays which do not depend on equality of moves are preserved by this map. These include length, depth of moves and visibility. However, notions that depend on equality of moves need not be preserved in any way. For example, this operation can create innocence (see Fig. 2) or break it (see Fig. 3). The interest of such a strong simplification comes from the fact that the hypotheses of our finiteness theorem do not include any assumption on the equality of moves, thus it suffices to consider plays on I_ω.

2.3. Pointer structures

As hinted above, it will suffice to state and prove a finiteness result for plays on the pure arena I_ω. However, these objects are relatively simple and do not require all the background and vocabulary of game semantics. Moreover, as argued in the introduction, these objects also arise independently of game semantics, *e.g.* in the execution of abstract machines. Thus, we give an elementary abstract definition which is the only we will used in the sequel.

In what follows, $\equiv_2$ denotes the even/odd equivalence on integers and $\mathbb{N}^*$ will denote $\mathbb{N} \setminus \{0\}$. We shall take, as in [15], an axiomatization for the structure of pointers on plays. We represent a pointer structure by a function

$$\phi : \mathbb{N} \longrightarrow \mathbb{N} \cup \{\bot\}.$$

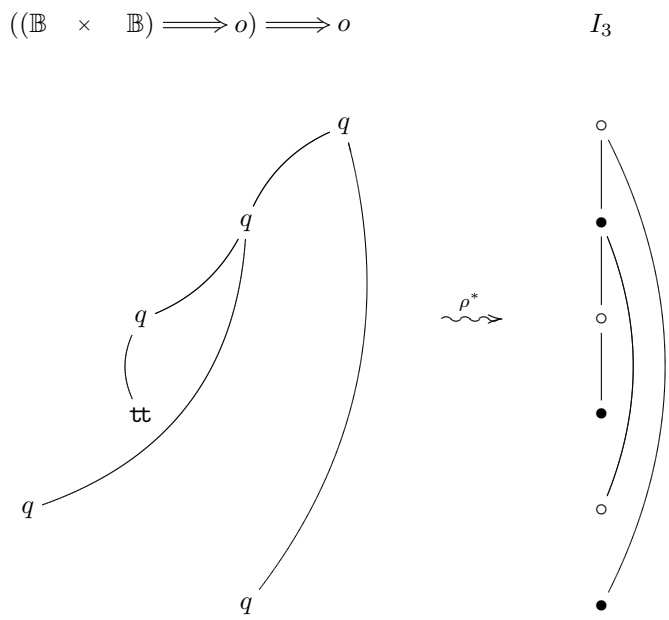


Figure 3: How collapsing breaks innocence

To each integer we associate the index of the move it points to; or $\perp$ if the play is already finished. We have the following axioms on pointer structures that formalize this intuition:

- Play finished: for all $i \in \mathbb{N}$, if $\phi(i) = \perp$ then, for all $j > i$, $\phi(j) = \perp$
- Pointing back: for all $i \in \mathbb{N}^*$, if $\phi(i) \neq \perp$ then $\phi(i) < i$
- Inversion of parity: for all $i \in \mathbb{N}^*$, $i \not\equiv_2 \phi(i)$
- Zero: $\phi(0) = 0$

The first axiom allows pointer structures to represent either finite or infinite plays. If ϕ is a pointer structure, its *domain*, denoted $\mathcal{D}_\phi$, is defined as $\{1, \ldots, k\}$ if k is the smallest integer such that $\phi(k+1) = \perp$; and as $\mathbb{N}$ if no such k exists. We say that a pointer structure ϕ is *finite* if $\mathcal{D}_\phi$ is finite.

The value of any pointer function at 0 is supposed to be 0, but it could instead be any other dummy value. Note in passing that, unlike the usual convention, even integers correspond here to Opponent moves and odd integers to Player moves. This is because it seems natural for 0 to represent the initial Opponent move. We write Ptr be the set of pointer structures.

Pointer structures allow us to define *views*. The definitions (but not the concepts) differ slightly from the usual ones.

Definition 1. *Let ϕ be a pointer structure. Its* view function $\ulcorner\phi\lrcorner : \mathcal{D}_\phi \to \mathcal{P}(\mathbb{N})$ *and* coview function $\llcorner\phi\urcorner : \mathcal{D}_\phi \to \mathcal{P}(\mathbb{N})$ *are computed in the following mutually recursive way:*

- $\ulcorner\phi\lrcorner(0) = \llcorner\phi\urcorner(0) = \{0\}$
- $\ulcorner\phi\lrcorner(i+1) = \{i+1\} \cup \llcorner\phi\urcorner(i)$
- $\llcorner\phi\urcorner(i) = \{i\} \cup \ulcorner\phi\lrcorner(\phi(i))$

Of course, the usual notions of P-view and O-view can easily be recovered as follows:

$$\ulcorner\phi\urcorner(i) = \begin{cases} \ulcorner\phi\lrcorner(i) & \text{if } i \text{ is odd} \\ \llcorner\phi\urcorner(i) & \text{if } i \text{ is even} \end{cases}$$

$$\llcorner\phi\lrcorner(i) = \begin{cases} \ulcorner\phi\lrcorner(i) & \text{if } i \text{ is even} \\ \llcorner\phi\urcorner(i) & \text{if } i \text{ is odd} \end{cases}$$

However, we prefer the first formulation because it allows, in the subsequent development, to reason about views without needing to make the parities of moves explicit. For example, a *visible pointer structure* is just a pointer structure ϕ satisfying, for all $i \in \mathcal{D}_\phi$, that $\phi(i) \in \ulcorner\phi\lrcorner(i)$; and an *ultimately visible pointer structure* is a pointer structure where there exists an $N \in \mathcal{D}_\phi$ such that, for all $i \in \mathcal{D}_\phi$ such that $i \geq N$, $\phi(i) \in \ulcorner\phi\lrcorner(i)$.

2.4. Finiteness theorems

Now, we have the vocabulary to state our main theorem. The reader should bear in mind that no proof will be given for the moment, the purpose of this section being to give and study equivalences between different statements of the theorem. From Section 3, the paper will be devoted to proving it.

Theorem 1 (Weak finiteness theorem)**.** *Let ϕ be an ultimately visible pointer structure. The following statements are equivalent.*

(i) there exists an $M \in \mathbb{N}$ such that, for all $i \in \mathcal{D}_\phi$, $|\ulcorner\phi\lrcorner(i)| \leq M$;

(ii) ϕ is finite.

Clearly, if ϕ is finite, its views have finite length. The converse is non-trivial and, as we will see, is similar in a very precise way to normalization of linear head reduction for simply-typed λ-calculus.

It is worth noticing that the finiteness of views can be decomposed into two complementary conditions. We call the *depth* of ϕ the longest pointer chain which can be extracted from ϕ:

$$\begin{aligned} d_\phi(0) &= 0 \\ d_\phi(i) &= 1 + d_\phi(\phi(i)) \\ depth(\phi) &= \sup_{i \in \mathcal{D}_\phi} d_\phi(i) \end{aligned}$$

In a similar way, we define a *fork* on ϕ to be a pair $(\psi_0, (\psi_i)_{i \in I})$ such that for all $i \in I$, $\phi(\psi_i) = \psi_0$. A fork is *conscious* iff, for all $i < j \in I$, $\psi_i \in \ulcorner \phi \urcorner(\psi_j)$: there is a sequence $\psi_1, \ldots, \psi_{|I|}$ of moves of the same polarity, all of which point to the same move ψ_0, and each of the ψ_j see all the previous ψ_i. The *size* of a fork $(\psi_0, (\psi_i)_{i \in I})$ is simply the cardinal of I. The *memory* of ϕ is then defined as the supremum of all $n \in \mathbb{N}$ such that ϕ admits a conscious fork of size n.

We have the following equivalence. Note that, unlike the other statements of this section, it does not depend on Theorem 1.

Proposition 2. *The two following statements are equivalent:*

(i) there exists an $M \in \mathbb{N}$ such that, for all $i \in \mathcal{D}_\phi$, $|\ulcorner \phi \urcorner(i)| \leq M$;

(ii) ϕ has finite depth and *finite memory.*

Proof. (i) $\Rightarrow$ (ii). Let $M \in \mathbb{N}$ be the bound on the size of views. Then neither pointer chains nor conscious forks can have size greater than M since this would create a view of size greater than M.
(ii) $\Rightarrow$ (i). Let d be the depth of ϕ and m its memory. Let M be the maximum size of a tree of depth d with branching degree m, which is always finite[3]. Then no view in ϕ is greater than M, otherwise the underlying tree given by the pointer structure on this view would have either a branching degree greater than m (which would break the memory condition) or a depth greater than d (which would break the depth condition). □

As an immediate corollary of this equivalence, we have the following alternative statement of Theorem 1.

Corollary 1. *Let ϕ be an ultimately visible pointer structure. If ϕ has finite depth and finite memory then ϕ is finite.*

The interest of this reformulation appears when we work with finite arenas. The structure of the arena then automatically restricts the depths of plays, so divergent plays must have infinite memory. In this case, restricting to strategies with finite memory therefore yields a category of games and total strategies.

In [19], Hyland introduced the *principle of justice*, to ban strategies which are *time wasting*, *i.e.* always repeating the same move while being aware of doing so. Apart from the fact that, in pointer structures, all moves at the same level are considered the same, this is really the same as restricting to strategies with finite memory. Hyland also stated a *compactness theorem*, saying that innocent strategies satisfying the principle of justice are finite (in the sense of having a finite view function). This is our Proposition 2: in finite arenas, where depth is bounded, it suffices to satisfy the memory condition to get a bound on the size of views. And of course, in finite arenas, it is equivalent for an innocent strategy to have bounded views and to have a finite view function.

Let us finally mention a third equivalent formulation of Theorem 1:

[3] We even have $M = \frac{m^{d+1}-1}{m-1}$

Proposition 3. *Let ϕ be an ultimately visible pointer structure with finite depth. If ϕ has forks of arbitrary size then it has conscious forks of arbitrary size.*

Proof. Let ϕ be an ultimately visible pointer structure with a finite depth, and suppose that ϕ has a fork of arbitrary size. Then, it is in particular infinite. By Corollary 1, it must have infinite memory, *i.e.* conscious forks of arbitrary size. Conversely, the present statement implies Corollary 1: if ϕ has finite depth and finite memory, it has only conscious forks of bounded size. So it only has forks of bounded size, thus it must be finite by König's lemma. □

Theorem 1 is sufficient for most cases. It allows to show the preservation of totality for a large class of strategies including the total and finite innocent strategies, but also any visible nondeterministic strategies whose views always remain of bounded size. In fact, we can do even better. In section 4, we prove the following generalization of the weak finiteness theorem:

Theorem 2 (Strong finiteness theorem)**.** *Let ϕ be an ultimately visible pointer structure with no infinite pointer chain. If ϕ has forks of arbitrary size then it has an* infinite *conscious fork.*

This theorem applies to a further class of innocent strategies, that we call *noetherian*, which can have views of unbounded length but no "infinite" views, *i.e.* no infinite strictly increasing sequence of views. For example, the strategy of type `nat -> com -> com` interpreting

```
let rec repeat n c =
  let z=n in
    if z=0 then () else (c; repeat (z-1) c)
```

is noetherian: it begins by asking for an n then calls its argument c exactly n times before converging. This strategy normalizes against any n (provided c converges of course) since none of its traces features an infinite conscious fork.

More generally, we show in Section 5 that the total noetherian strategies are closed under composition. Unlike the case of total bounded strategies, which only needs the weak finiteness theorem, this result crucially depends on the strong finiteness theorem, as the above example suggests.

3. Pointer structures and λ-calculus

It has long been known [8, 9] that there is a tight connection between the hyper-lazy reduction strategy called linear head reduction for λ-calculus and the legal plays produced by innocent interaction. This connection establishes that pointing strings can alternatively be seen as traces of the execution of an abstract machine called the PAM (Pointer Abstract Machine) that computes linear head reduction. We are not going to recall the details of the PAM in this paper because it is very technical and we do not really need it. However, this connection suggests that we take inspiration from the normalization proof of simply-typed λ-calculus to build a finiteness theorem for pointer structures.

To begin with, what kind of λ-calculus would pointer structures correspond to? Since the only arenas are the I_n, we suspect the types would be as follows:

$$\begin{aligned} \underline{0} &= o \\ \underline{n+1} &= \underline{n} \to o \end{aligned}$$

Thus, all typed λ-terms would be unary; only one abstraction and application would be allowed at each level. A complication is that, as highlighted above, our pointer structures may witness non-innocent behaviour. Hence, the execution of these λ-terms must be able to cause ruptures of innocence. The simplest way to do that is to add a nondeterministic choice operator, as we do not care about completeness.

The resulting typing rules are the following:

Unary Λ

$$\frac{}{\Gamma, x : \underline{k} \vdash x : \underline{k}}\, ax \qquad \frac{}{\Gamma \vdash \maltese : \underline{k}}\, dai$$

$$\frac{\Gamma, x : \underline{k} \vdash M : \underline{0}}{\Gamma \vdash \lambda x.M : \underline{k+1}}\, lam \qquad \frac{\Gamma \vdash M : \underline{k+1} \qquad \Gamma \vdash N : \underline{k}}{\Gamma \vdash MN : \underline{0}}\, app$$

$$\frac{\Gamma \vdash M : \underline{k} \qquad \Gamma \vdash N : \underline{k}}{\Gamma \vdash M + N : \underline{k}}\, sum$$

This calculus is equipped with the following nondeterministic reduction rules. We will only be interested in *head reduction*, *i.e.* we only perform leftmost reduction.

$$\begin{array}{rcll} (\lambda x.M)N & \rightsquigarrow & M[N/x] & (\beta) \\ \maltese N & \rightsquigarrow & \maltese & (\maltese) \\ M + N & \rightsquigarrow & M & (+_l) \\ M + N & \rightsquigarrow & N & (+_r) \end{array}$$

As this is a simply-typed λ-calculus, it is simple to prove normalization of its head reduction, *e.g.* with Kleene realizability. For the sake of completeness, let us sketch the proof. Let Λ denote the set of *closed* terms.

Definition 2 (Realizability)**.** *We define a relation $\Vdash \subseteq \Lambda \times \mathbb{N}$:*

- $M \Vdash 0 \Leftrightarrow M \rightsquigarrow^* \maltese$
- $M \Vdash k+1 \Leftrightarrow \forall N \Vdash k,\ MN \rightsquigarrow^* \maltese$.

Lemma 1 (Adequacy)**.** *Suppose $x_1 : \underline{n_1}, \ldots, x_p : \underline{n_p} \vdash M : \underline{k}$. Then for all $N_1 \Vdash n_1, \ldots, N_p \Vdash n_p$, $M[N_1/x_1, \ldots, N_p/x_p] \Vdash k$.*

This is proved by induction on the derivation tree of M. Normalization of head reduction is then an easy consequence: we prove that $\maltese$ is a realizer of all k. Thus, since any M such that we can derive $\vdash M : \underline{k}$ satisfies $M \Vdash k$, either $k = 0$ and $M \rightsquigarrow^* \maltese$ and hence the reduction terminates; or $M\maltese \Vdash 0 \rightsquigarrow^* \maltese$, but an infinite reduction sequence for M would also be an infinite reduction sequence for $M\maltese$, hence the reduction sequence of M terminates.

3.1. Linear head reduction

The above argument seems incompatible with game semantics since innocent strategies correspond to Böhm trees, *i.e.* λ-terms in η-long β-normal form, whereas the mechanism of substitution necessary to express β-reduction does not preserve the Böhm tree structure. Nevertheless, in this subsection we show how to deduce normalization of linear head reduction from that of head reduction. This enables us to use the above result to give a (syntactic) proof of the finiteness of pointer structures.

As pointed out at the beginning of this section, pointer structures are a description of the *linear head reduction sequence* of our unary λ-terms. As this is not so well known, we recall it here. However, this section is rather informal and mainly serves to introduce the ideas behind the normalization proof for pointer structures, so we elide treatment of $\maltese$ and nondeterministic choice. The interested reader can find an in-depth description in [9].

In what follows, we need to carefully distinguish the notions of *variables* and of particular *occurrences* of those variables in a term. We choose the following convention: variables are denoted by x and y while occurrences of x (resp. y) are denoted by $x_0, x_1, \dots$ (resp. $y_0, y_1, \dots$). The *head occurrence* of a λ-term M is the variable occurrence in head position, determined as follows:

$$\begin{aligned} hoc(MN) &= hoc(M) \\ hoc(\lambda y.M) &= hoc(M) \\ hoc(x_i) &= x_i \end{aligned}$$

Suppose x_0 is the head occurrence of M, and is an occurrence of a variable x bound in M. This means that there is a subterm T of M of the form $\lambda x. \dots$, which has also x_0 as head occurrence. Now, suppose T has an argument N in M, which means that TN is a subterm of M. Note that if it is the case, it can have only one argument because the calculus is unary. We call N the *argument* of x_0 and call the pair (x_0, N) a linear head redex. Reducing this linear head redex then consists of substituting the occurrence x_0 by N, leaving all other occurrences of x and also the argument subterm N unchanged. In summary:

$$\dots(\lambda x.(\dots(x_0 \dots))N \dots \quad \rightsquigarrow \quad \dots(\lambda x.(\dots(N \dots))N \dots$$

The reader may note that, in this setting, linear head redexes always correspond to usual β-redexes, even though reduction is not β-reduction. This is not true in general for linear head reduction, where redexes are only defined up to σ-equivalence [9], but *is* true here because our types and terms are simpler.

This reduction strategy can easily be proved normalizing by reducing it to head β-reduction. Here is an elegant way to do that which we could not find in the literature.

Proposition 4. *Let M be an unary λ-term such that $\Gamma \vdash M : k$ is provable. If all head β-reduction sequences of M are finite then all linear head reduction sequences of M are finite.*

Proof. Consider the language of λ-terms temporarily extended by a term $\boxed{T}$, for each term T, with the reduction rule

$$\boxed{T} \rightsquigarrow_{pop} T$$

that applies whenever $\boxed{T}$ appears in head position. Suppose there is $\vdash M : k$ together with an infinite reduction sequence

$$M \rightsquigarrow_{LHR} M_1 \rightsquigarrow_{LHR} M_2 \rightsquigarrow_{LHR} M_3 \rightsquigarrow_{LHR} \cdots$$

We reason by induction on the length of the head β-reduction chain of M which is necessarily finite since M is typed.

We begin by decomposing β-reduction into two steps:

1. *Substitution ($\beta\Box$)* $(\lambda x.M)N \rightsquigarrow_{\beta\Box} M[\boxed{N}/x]$
2. *Unboxing. ($\Box^{-1}$)* In one step, we remove all boxes in M.

Suppose that $M \rightsquigarrow_{\beta\Box} M'$. We can easily transport the above infinite reduction sequence as:

$$\begin{array}{ccccccccc} M & \xrightarrow{LHR} & M_1 & \xrightarrow{LHR} & M_2 & \xrightarrow{LHR} & M_3 & \xrightarrow{LHR} & \cdots \\ \downarrow{\scriptstyle \beta\Box} & & & & & & & & \\ M' & \xrightarrow{LHR\vee pop} & M'_1 & \xrightarrow{LHR\vee pop} & M'_2 & \xrightarrow{LHR\vee pop} & M'_3 & \xrightarrow{LHR\vee pop} & \cdots \end{array}$$

The idea is that at each time we reduce, in the upper infinite chain, the same redex that we reduced initially for $M \rightsquigarrow_{\beta\Box} M'$, we instead open the box in head position.

Now we just need to remove the unnecessary steps in the lower reduction sequence:

$$\begin{array}{ccccccccc} M' & \xrightarrow{LHR\vee pop} & M'_1 & \xrightarrow{LHR\vee pop} & M'_2 & \xrightarrow{LHR\vee pop} & M'_3 & \xrightarrow{LHR\vee pop} & \cdots \\ \downarrow{\scriptstyle \Box^{-1}} & & & & & & & & \\ M'' & \xrightarrow{LHR} & M''_{\sigma(1)} & \xrightarrow{LHR} & M''_{\sigma(2)} & \xrightarrow{LHR} & M''_{\sigma(3)} & \xrightarrow{LHR} & \cdots \end{array}$$

where $\sigma : \mathbb{N} \to \mathbb{N}$ is a strictly increasing function, used to conceal the now unnecessary unboxing steps.

The point is that, if the reduction sequence of M' is infinite, so is that of M''. Indeed, this is obvious if the number of times the *pop* rule is used in the reduction sequence of M' is finite; but if that is infinite, the number of linear head reductions must be infinite as well—otherwise, the reduction sequence would have an infinite tail of *pop*-reductions which is absurd since a *pop*-reduction never creates a *pop*-redex. So the number of linear head reductions is infinite in the reduction sequence of M', but these are precisely the reductions that remain in the chain of M''.

However, we know by the induction hypothesis that the linear head reduction chain of M'' is finite, since $M \rightsquigarrow_\beta M''$. Note that the base case is trivial since a normal form for head β-reduction is also a normal form for linear head reduction. This concludes the proof. □

Sadly, we cannot transport this argument to pointer structures because of the dependency on β-reduction. A natural workaround is to use the PAM to relate formally normalization of linear head reduction to finiteness of pointer structures. We only sketch the basic idea and, as such, we remain very informal. The interested reader should see [9] for the precise definitions.

The PAM is a device which, given any λ-term, simulates its linear head reduction sequence by keeping track at each step i of three different kinds of data: the current *head occurrence* h_i, its *argument* A_i (in the sense introduced above), and a pointer p_i back to the index of the reduction step i' when $A_{i'}$ began with an abstraction of the variable of which h_i is an occurrence. It was proved in [8] that, if M and N are two λ-terms in η-long β-normal form, the (p_i) sequence obtained by execution of the PAM on MN corresponds exactly to the pointers one witnesses in $[\![M]\!]||[\![N]\!]$, where $[\![\cdot]\!]$ computes the usual interpretation of λ-terms in HO games semantics and $||$ denotes parallel composition.

It is possible to define a *reversed* version of the PAM which takes as input a pointer structure ϕ and produces two typed λ-terms, M and N, such that ϕ is the (p_i) sequence of the execution of MN by the usual PAM. Clearly, this machine stops if, and only if, ϕ is finite. Moreover, it can be proved (without induction) that the machine stops if, and only if, the linear head reduction sequence of MN is finite. So we have a very strong equivalence between termination of linear head reduction and finiteness of pointer structures.[4]

However, the goal here is to provide a semantic finiteness proof for pointer structures, if possible getting rid of any dependence on the calculus. Thus we must not look for formal connections of pointer structures with linear head reduction, but rather for direct normalization arguments in pointer structures, possibly influenced by the calculus. Hence it is natural to search for direct normalization arguments for linear head reduction, in no way relying on β-reduction.

[4]The details can be found here: `http://www.pps.jussieu.fr/~pclairam/games/rpam.pdf`

3.2. Realizability proof of normalization

It is not so easy to show, independently of β-reduction, that linear head reduction normalizes. Let us try to explain why. Suppose M and N are two closed λ-terms in η-long and β-normal form. Necessarily, M has the form $M = (\lambda x.xM')$. We then have the following (linear head) reduction for MN:

$$(\lambda x.xM')N \rightsquigarrow (\lambda x.NM')N$$

Note that no linear head reduction can affect later the general form of the term: it will remain for the rest of the reduction of the form $(\lambda x.T)N$, for a given T. Thus, all that is needed to compute the reduction sequence of MN is the reduction sequence of NM'. However, M' is no longer necessarily closed: it can contain free occurrences of x. It is easy to show that, in fact, a term of the reduction sequence of MN always has the form

$$(\lambda x.(\lambda y.(\lambda z.(\dots(UV)\dots)\dots)M_z)M_y)M_x$$

which is a sequence of abstractions, followed by the application of two terms, followed by a sequence of applications. A useful remark is that, since all these abstractions/applications remain in the same place during the reduction, the above term could be denoted by a pair (UV, ρ) where $\rho = \{x \mapsto M_x, y \mapsto M_y, z \mapsto M_z, \dots\}$ is an environment. This is exactly what the *Krivine machine with environment* [9] does; this machine is typically stated to be correct with respect to head β-reduction, but in fact computes linear head reduction.

Anyway, the point about normalization is that we are led to consider interactions of open terms, which prevents us from using the same easy realizability argument as for head β-reduction. Instead of terms we need to consider *closures*, *i.e.* terms with their run-time environment.

To successfully take inspiration from the above remarks and formulate a realizability proof in pointer structures, we need to find a counterpart to the following notions:

Types. No surprise here, a type is still denoted by an integer. Intuitively, it is the depth of the underlying arena.

Closed terms. Starting from a pointer structure ϕ, we need to find the equivalent of two unary λ-terms with nondeterministic choice whose possible interactions include ϕ. Using intensively the nondeterministic feature of our language, we just use integers for our terms, to be understood as the maximal length of their Böhm tree. Formally, we define:

Definition 3. *A* closed interaction between integers k and p *is a visible pointer structure ϕ such that:*

$$\forall i \in \mathcal{D}_\phi,\ |\ulcorner\phi\urcorner(i)| \leq k \ \wedge\ |\llcorner\phi\lrcorner(i)| \leq p+1$$

The set of all closed interactions between k and p is denoted by $k \star p$.

Contexts. What is an interaction between two open terms in a given context? This happens to have a simple answer in this setting: since this situation arises naturally when using linear head reduction after a few reduction steps, we will consider pointer structures with a specified node, called the *starting move*. The part of the pointer structure before the starting move is considered to be the *history* of the play. Pointing to a move in the history is making a call to the environment.

Generalized terms. We need to extend our notion of term to this generalized kind of interaction. Take a pointer structure ϕ and its starting move $i \in \mathcal{D}_\phi$. What are the terms interacting in the play beginning at i ? By extension of the previous definition, they should be the sizes of the views starting at any move of ϕ whose views include i, and stopping whenever i is reached:

Definition 4. *Let ϕ be a pointer structure, and let $i \in \mathcal{D}_\phi$. The* maximal view from i*, denoted by $\overline{\ulcorner \phi \lrcorner}(i)$, is defined as follows:*

$$\overline{\ulcorner \phi \lrcorner}(i) = \max \left(\begin{array}{c} \\ \max\limits_{\substack{i \in \ulcorner \phi \lrcorner(j) \\ i \equiv_2 j}} |\ulcorner \phi \lrcorner(j)| - |\ulcorner \phi \lrcorner(i)| + 1, \quad \max\limits_{\substack{i \in \llcorner \phi \urcorner(j) \\ i \not\equiv_2 j}} |\llcorner \phi \urcorner(j)| - |\ulcorner \phi \lrcorner(i)| + 1 \\ \\ \end{array} \right)$$

Similarly, the maximal coview from i *is defined by:*

$$\overline{\llcorner \phi \urcorner}(i) = \max \left(\begin{array}{c} \\ \max\limits_{\substack{i \in \llcorner \phi \urcorner(j) \\ i \equiv_2 j}} |\llcorner \phi \urcorner(j)| - |\llcorner \phi \urcorner(i)| + 1, \quad \max\limits_{\substack{i \in \ulcorner \phi \lrcorner(j) \\ i \not\equiv_2 j}} |\ulcorner \phi \lrcorner(j)| - |\llcorner \phi \urcorner(i)| + 1 \\ \\ \end{array} \right)$$

The maximal view and coview from i can really be understood as the terms interacting in the subplay starting with i, with the history $\phi_{<i}$.

Closures. The above definition only partially describes the agents interacting in an open context. We need to give a description of their environment with them. An *agent* will be an integer n (still representing a Böhm tree size) and a finite list of agents $a_1, \ldots, a_p$. The idea is that, as long as the current move is hereditarily justified by the starting move, the play obeys the restrictions of n but, as soon as n "calls" the jth element of its view, the play will obey the restrictions of the agent a_j.

Definition 5. *An* agent *is an integer n and a tuple of agents $a_1, \ldots, a_p$, denoted by $n[a_1, \ldots, a_p]$.*

The set of all agents is denoted by $\mathcal{A}$. This definition is well-founded since we can build an agent from a 0-tuple—in which case, we drop the brackets and write n as shorthand for $n[]$.

An alternative explanation of this definition is that it provides a way to speak of strategies that are not in β-normal form. Indeed, $n[a_1, \ldots, a_p]$ can be understood as a β-normal term with p free variables, and performing the corresponding substitutions with $a_1, \ldots, a_p$ *without renormalizing the resulting term.* This is rather more subtle than the usual interpretation of substitution in game semantics—parallel composition plus hiding—which implicitly renormalizes the term to a (possibly infinite) Böhm tree—infinite in the case that the resulting term cannot be normalized. This interpretation of the definition of agents justifies the chosen notation $n[a_1, \ldots, a_p]$, to be understood intuitively as the result of the substitution of n by $a_1, \ldots, a_p$.

It remains to define how agents interact. The idea should now seem natural: the result of the interaction of two agents a and b is a pointer structure ϕ along with its starting move i, such that views are compatible with the size of a and b and such that calls to the history are compatible with the closure parts of a and b.

Definition 6. *Let $a = n[a_1, \ldots, a_p]$ be an agent. The set of* traces *of a, denoted by $Tr(a)$, is the set of pairs (ϕ, i), with $i \in \mathcal{D}_\phi$, such that:*

- $\overline{\ulcorner \phi \urcorner}(i) \leq n$
- $|\{j \in \llcorner \phi \urcorner(i) \mid (j \equiv_2 i) \wedge (j < i)\}| = p$ *and if* $\{j_1, \ldots, j_p\} = \{j \in \llcorner \phi \urcorner(i) \mid (j \equiv_2 i) \wedge (j < i)\}$ *(ordered by $<$) then for all $m \in \{1, \ldots, p\}$, for all $x \in \phi^{-1}(j_m)$ such that $x \geq i$, $(\phi, x) \in Tr(a_m)$.*
- *ϕ is visible after i*

Definition 7. *If n and p are agents then the set of interactions between n and p, denoted $n \star p$, is defined as follows:*

$$n \star p = \{(\phi, i) \in Ptr \times \mathbb{N} \mid (\phi, i) \in Tr(n),\ \forall x \in \phi^{-1}(i),\ (\phi, x) \in Tr(p)\}$$

We have by now gathered together all the necessary material to build a normalization proof for pointer structures inspired by the realizability proof of the beginning of Section 3.

Definition 8 (Realizability)**.** *We define the realizability relation $\Vdash \subseteq \mathcal{A} \times \mathbb{N}$ as follows:*

- $n \Vdash 0 \Leftrightarrow n \leq 1$
- $n \Vdash d + 1 \Leftrightarrow \forall m \Vdash d,\ \forall (\phi, i) \in n \star m,\ \phi$ *is finite.*

Note how agents have assumed the role of realizers played by *closed* λ-terms in the previous argument.

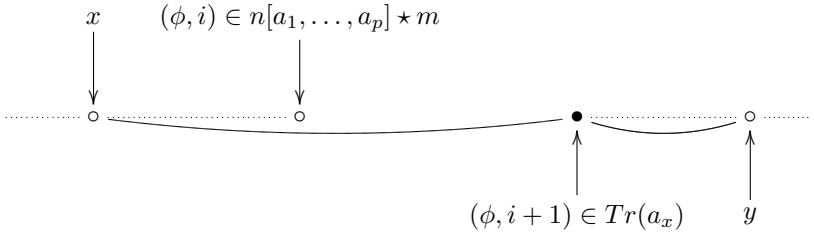
Lemma 2 (Adequacy)**.** *For all $n \in \mathbb{N}$, if we have $d_1, \ldots, d_p, d \geq n$ and $a_1 \Vdash d_1, \ldots, a_p \Vdash d_p$, then $n[a_1, \ldots, a_p] \Vdash d$.*

Proof. By induction on n.

If $n = 0$ then $Tr(n[a_1, \ldots, a_p])$ is always empty. If $n = 1$ and $(\phi, i) \in Tr(n[a_1, \ldots, a_p])$ then necessarily $\phi(i+1) = \bot$ and ϕ is finite.

If $n > 1$, suppose we have $a_1 \Vdash d_1, \ldots, a_p \Vdash d_p$ satisfying the hypotheses of the lemma, $d \in \mathbb{N}$ such that $d + 1 \geq n$ and $m \Vdash d$. Let also $(\phi, i) \in n[a_1, \ldots, a_p] \star m$ and consider $\phi(i+1)$. If it is equal to $\bot$ then ϕ is finite. Otherwise, we have to prove that $(\phi, i+1) \in q \star (n-2)[a_1, \ldots, a_p, m]$, for q a realizer of some integer l, in order to conclude by the induction hypothesis.

Let $\{j_1, \ldots, j_{p+1}\} = \{j \in \ulcorner \phi \urcorner(i+1) \mid j \not\equiv_2 i+1\}$. By visibility, there is $x \in \{1, \ldots, p+1\}$ such that $\phi(i+1) = j_x$. Since $(\phi, i) \in n[a_1, \ldots, a_p] \star m$, either $x \leq p$ and $(\phi, i+1) \in Tr(a_x)$ or $x = p+1$ and $(\phi, i+1) \in Tr(m)$. Take now $y \in \mathcal{D}_\phi$ such that $\phi(y) = i+1$. We need to show that $(\phi, y) \in Tr((n-2)[a_1, \ldots, a_p, m])$. Let us illustrate the current state of ϕ:



The first point to check is that $\overline{\ulcorner \phi \urcorner}(y) \leq n-2$. This is clear since by definition $\overline{\ulcorner \phi \urcorner}(i) \geq \overline{\ulcorner \phi \urcorner}(y) + 2$. Now:

$$\begin{aligned} \llcorner \phi \urcorner(y) &= \{y\} \cup \ulcorner \phi \urcorner(i+1) \\ &= \{y, i+1\} \cup \llcorner \phi \urcorner(i) \end{aligned}$$

So $\{j \in \llcorner \phi \urcorner(y) \mid (j < y) \wedge (j \equiv_2 y)\} = \{j_1, \ldots, j_p, i\}$. Unfolding the definition of $(\phi, i) \in n[a_1, \ldots, a_p] \star m$ ensures directly that $(\phi, y) \in Tr((n-2)[a_1, \ldots, a_p, m])$. Thus $(\phi, i+1) \in q \star (n-2)[a_1, \ldots, a_p, m]$, with either $q = m$ or $q = a_x$ for some $x \in \{1, \ldots, p\}$. In either case there is, by hypothesis, some $l \in \mathbb{N}$ such that $q \Vdash l$, with $l \geq min(d_1, \ldots, d_p, d) \geq n-1$. Hence $l - 1 \geq n-2$ and, by induction hypothesis, $(n-2)[a_1, \ldots, a_p, m] \Vdash l - 1$ and ϕ is finite. □

With the adequacy lemma in place, we can already deduce a normalization theorem for *visible* pointer structures. Indeed, suppose that ϕ is a visible pointer structure whose views are bounded by M? We know that $\phi \in M \star M + 1$ and the adequacy lemma ensures that $M \Vdash M + 2$ and $M + 1 \Vdash M + 1$. Hence ϕ is finite.

The situation for pointer structures which are only *ultimately* visible is slightly more subtle since, if i is the move after which ϕ is visible, we have to recover the agents interacting at i, *i.e.* with a non-empty environment. Thus, in the general case, an additional property is required to rebuild them. This is the object of our next lemma.

Lemma 3. *Let ϕ be a pointer structure such that views in ϕ are bounded by $M \in \mathbb{N}$, and $i \in \mathcal{D}_\phi$ such that ϕ is visible after i. Then*

(i) We are able to build an agent $a_i \in \mathcal{A}$ such that $(\phi, i) \in Tr(a_i)$. Moreover the resulting agent only depends on $\phi(i)$,

(ii) We are able to build another agent $b_i \in \mathcal{A}$ such that $(\phi, i) \in a_i \star b_i$,

(iii) For all $d \geq M$, $a_i \Vdash d$ and $b_i \Vdash d$.

Proof. (i). By induction on $\phi(i)$. If $\phi(i) = 0$, then $(\phi, i) \in Tr(M)$. Otherwise define:

$$\{j_1, \ldots, j_p\} = \{x \in \llcorner \phi \urcorner(i) \mid x \equiv_2 i \wedge x < i\}.$$

Indeed, $\{x \in \llcorner \phi \urcorner(i) \mid x \equiv_2 i \wedge x < i\}$ cannot be empty if $\phi(i) \neq 0$ since it contains at least $\phi(i) - 1$. Now, for each j_x and for each $y > i \in \mathcal{D}_\phi$ such that $\phi(y) = j_x$, we have by induction hypothesis an agent a_x such that $(\phi, y) \in Tr(a_x)$. Moreover, the induction hypothesis also ensures that a_x does not depend on the choice of y. Thus,

$$(\phi, i) \in Tr(M[a_1, \ldots, a_p]).$$

It remains to prove that this agent does not depend on i. It suffices to remark that if $i' \in \mathcal{D}_\phi$ is such that ϕ is visible after i' and $\phi(i) = \phi(i')$, we also have

$$\{x \in \llcorner \phi \urcorner(i) \mid x \equiv_2 i \wedge x < i\} = \{x \in \llcorner \phi \urcorner(i') \mid x \equiv_2 i' \wedge x < i'\}.$$

Thus, exactly the same argument applies and $(\phi, i') \in Tr(M[a_1, \ldots, a_p])$.

(ii). Either $i = 0$, in which case $(\phi, i) \in M \star M$. Otherwise, we compute

$$\{j_1, \ldots, j_q\} = \{x \in \ulcorner \phi \lrcorner(i) \mid x \not\equiv_2 i\}.$$

For each j_x, *(i)* provides an agent b_x such that, for all $y > i$ such that $\phi(y) = j_x$, $(\phi, y) \in Tr(b_x)$, and this agent does not depend on the choice of y. Thus we get $b_1, \ldots, b_q$ such that $(\phi, i) \in a_i \star M[b_1, \ldots, b_q]$.

(iii) Straightforward induction on the definition of the agents built, using the adequacy lemma at each step. □

We can now prove the weak finiteness theorem for pointer structures.

Theorem 1 (revisited). *Let ϕ be an ultimately visible pointer structure. The following propositions are equivalent.*

(i) $\exists M \in \mathbb{N},\ \forall i \in \mathcal{D}_\phi,\ |\ulcorner \phi \lrcorner(i)| \leq M$

(ii) ϕ is finite.

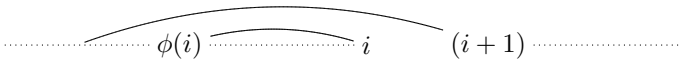
Proof. The only non-trivial direction is *(i)* ⇒ *(ii)*. Let $M \in \mathbb{N}$ be the bound on the size of views. By Lemma 3, we are able to build $a, b \in \mathcal{A}$ such that $(\phi, i) \in a \star b$, with $a \Vdash M + 1$ and $b \Vdash M$. Thus ϕ is finite. □

The reader may wonder why this proof only solves the "bounded views" formulation of the theorem and not the stronger "no infinite view" introduced at the end of Section 2. In fact, the limitation does not lie in the proof technique, but in the class of agents considered. To handle agents without bounded views, one would be forced to enrich the syntax of agents with an infinitary nondeterministic choice operation at each level of the Böhm trees, so that terms with unbounded views would be allowed—without the induction required to prove adequacy becoming non-well-founded. Some necessary definitions would then become very difficult to state, starting with the definition of $Tr(a)$. We believe the proof would lose a significant part of its explanatory aspect.

4. Alternative direct normalization proof

In this section, we present a different, more elementary proof of the finiteness theorem. It is slightly more general, but does not involve the interesting structure necessary to the realizability proof, and is also less insightful. The proof is similar to the "no infinite chattering" argument found in [6] which can be traced back to [5] which predates modern game semantics.

The starting point is a known property of visible plays. Suppose we have a visible pointer structure ϕ with a node $i \in \mathcal{D}_\phi$ such that nobody ever points to i:



Then the segment $\{\phi(i), \ldots, i\}$ is unreachable for the rest of the play: nobody will be able to point there. As a consequence, we can safely remove it without affecting views for the remaining part of the play. Thus, we get a subplay of ϕ where one node without successor has been removed. If one regards this erasing operation as a rewriting system on visible subsequences of ϕ, it is clear that, if ϕ is initially infinite, a normal form must be a subsequence of ϕ for which every move has a successor—which can exist only if ϕ has infinite depth. Thus, if we can show that this reduction terminates, the proof will be over. This is not obvious since there can be an arbitrary number of moves without successor in an infinite play, and this operation may also very well generate new such moves.

We can deal with this by the following argument. We define an order $\prec$ on moves of the same polarity by $x \prec y$ *iff* x *sees* y. The hypothesis of finiteness of views generalizes into the slightly weaker assumption that $\prec$ is well-founded. Consider then the induced lexicographic order $\prec^*$ on subsequences of ϕ. In the pattern above, we always have $i+1 \prec \phi(i)$, so the reduction is strictly decreasing for $\prec^*$. This is not yet enough: subsequences of ϕ can be infinite, hence $\prec^*$ need not be well-founded. However, an additional argument will allow us to construct directly a visible subsequence of ϕ which is minimal for $\prec^*$ and thus must be normal for the reduction sketched above, which will conclude the proof. Of course, the low-level character of pointer structures makes the formalization of this proof a little bit longer.

We first have to define, for a fixed ϕ, the set of its visible subsequences. For that purpose, we need some preliminary definitions.

Definition 9. *Let ϕ be a pointer structure, let $(u_i)_{i\in I} \in {\mathcal{D}_\phi}^I$ be a strictly increasing sequence of integers in the domain of ϕ (where I is an initial segment of $\mathbb{N}$). Then:*

- *u is a* subsequence *of ϕ iff*

$$\begin{cases} \text{for all } i \in I \text{ there is } j < i \text{ such that } u_j = \phi(u_i) \\ \text{for all } i+1 \in I,\ u_i \not\equiv_2 u_{i+1} \end{cases}$$

- *The* restriction *of ϕ to u, denoted ϕ_u, is defined as:*

$$\phi_u : n \mapsto \begin{cases} \bot & \text{if } n \notin I \\ m & \text{such that } u_m = \phi(u_n) \text{ otherwise.} \end{cases}$$

- *u is a* visible subsequence *of ϕ iff for all $i, j \in I$:*

$$i \in \ulcorner \phi_{u\lrcorner}(j) \Leftrightarrow u_i \in \ulcorner \phi_\lrcorner(u_j)$$

The set of visible subsequences of ϕ is denoted by $VSub(\phi)$.

Let us now introduce the orders $\prec$ and $\prec^*$.

Definition 10. *We define $\prec$ as follows: let $i, j \in \mathcal{D}_\phi$, then*

$$i \prec j \iff j \in \ulcorner \phi_\lrcorner(i) \wedge i \equiv_2 j \wedge i \neq j$$

The induced lexicographic order is $\prec^*$ on $VSub(\phi)$.

Lemma 4. *If ϕ is infinite and such that $\prec$ is well-founded, then $VSub(\phi)$, ordered by $\prec^*$, has an infinite element $(m_i^\phi)_{i\in I}$ which is minimal among infinite visible subsequences of ϕ.*

Proof. We build m^ϕ as follows :

- $m_0^\phi = 0$
- Consider the set of integers k for which there is an infinite visible subsequence of ϕ beginning by $m_0^\phi, \ldots, m_n^\phi, k$. The fact that this set is non-empty is an induction invariant, and it must admit a minimal element (for $\prec$) by well-foundedness of $\prec$. Let m_{n+1}^ϕ be this minimal element.

This defines $m^\phi \in VSub(\phi)$. Now if there was an infinite $u \in VSub(\phi)$ such that $u \prec^* m^\phi$, consider the least $n \in \mathbb{N}$ such that $m_n^\phi \neq u_n$. By definition of $\prec^*$, we must have then $u_n \prec m_n^\phi$, which is impossible by construction of m^ϕ. □

It remains to define the announced reduction and prove its compatibility with respect to $\prec^*$. First, the fundamental lemma on views which enables us to define the reduction:

Lemma 5. *Suppose ϕ is a pointer structure, take $i \in \mathcal{D}_\phi$ such that ϕ is visible after i, and take (u) a visible subsequence of ϕ. Then, for all $j > i$,*

$$\nexists k \in \{i{+}1, \ldots, j{-}1\},\ \phi(k) = i \implies \ulcorner\phi_\lrcorner(j) \cap \{x \mid x \not\equiv_2 j\} \cap \{\phi(i), \ldots, i{-}1\} = \emptyset$$

Proof. By induction. Let first j be $i+1$. Then:

$$\begin{array}{rcl} \ulcorner\phi_\lrcorner(i+1) & = & \{i+1\} \cup \llcorner\phi\urcorner(i) \\ & = & \{i+1, i, \phi(i)\} \cup \llcorner\phi\urcorner(\phi(i)-1) \end{array}$$

Therefore:

$$\ulcorner\phi_\lrcorner(i+1) \cap \{x \mid x \not\equiv_2 i+1\} \cap \{\phi(i), \ldots, i-1\} = \emptyset$$

Consider now $j > i+1$, and suppose $\nexists k \in \{i+1, \ldots, j-1\},\ \phi(k) = i$. Then:

$$\begin{array}{rcl} \ulcorner\phi_\lrcorner(j) \cap \{\phi(i), \ldots, i-1\} & = & (\{j\} \cup \llcorner\phi\urcorner(j-1)) \cap \{\phi(i), \ldots, i-1\} \\ & = & \llcorner\phi\urcorner(j-1) \cap \{\phi(i), \ldots, i-1\} \\ & = & \ulcorner\phi_\lrcorner(\phi(j-1)) \cap \{\phi(i), \ldots, i-1\} \end{array}$$

Since ϕ is supposed to be visible after i, we know that $\phi(j{-}1) \in \ulcorner\phi_\lrcorner(j{-}1)$. Thus the induction hypothesis ensures that $\phi(j-1) \notin \{\phi(i), \ldots, i{-}1\}$. Moreover, by assumption $\phi(j-1) \neq i$ thus $\phi(j-1) \notin \{\phi(i), \ldots, i\}$.

Therefore, two cases arise:

- Either $\phi(j-1) < \phi(i)$ and

$$\ulcorner\phi_\lrcorner(\phi(j-1)) \cap \{\phi(i), \ldots, i-1\} = \emptyset$$

- Or $\phi(j-1) \in \{i+1, \ldots, j-2\}$. In that case we know by induction hypothesis that:

$$\ulcorner\phi_\lrcorner(\phi(j-1)) \cap \{x \mid x \not\equiv_2 \phi(j-1)\} \cap \{\phi(i), \ldots, i-1\} = \emptyset$$

And since $\{x \mid x \not\equiv_2 \phi(j-1)\} = \{x \mid x \not\equiv_2 j\}$, we conclude that

$$\ulcorner\phi_\lrcorner(j) \cap \{x \mid x \not\equiv_2 j\} \cap \{\phi(i), \ldots, i-1\} = \emptyset \qquad \square$$

We deduce from this the desired reduction on visible subsequences of ϕ.

Lemma 6. *Let ϕ be a pointer structure, $(u_i)_{i \in I} \in VSub(\phi)$, $i \in I$ such that*

- *There is no $j \in I$ satisfying $\phi(u_j) = u_i$,*
- *ϕ is visible after $\phi(u_i)$,*

Then there exists $v \in VSub(\phi)$ such that $v \prec^* u$.

Proof. We first define v' as a subsequence of ϕ_u:

$$\begin{cases} v'_n = n & \text{if } n < \phi_u(i) \\ v'_n = n + i - \phi_u(i) + 1 & \text{otherwise.} \end{cases}$$

Lemma 5 applied to ϕ_u ensures that v' is a subsequence of ϕ_u. Indeed, take $v'_j > i$. Since nobody will ever point to i in ϕ_u, we have:

$$\ulcorner \phi_u \lrcorner(v'_j) \cap \{x \mid x \not\equiv_2 v'_j\} \cap \{\phi_u(i), \ldots, i-1\} = \emptyset$$

Thus $\phi_u(v'_j) \notin \{\phi_u(i), \ldots, i\}$ and by visibility there must be k such that $\phi_u(v'_j) = v'_k$.

Let us now check that v' is a visible subsequence of ϕ_u, that is:

$$j_1 \in \ulcorner \phi_{v'} \lrcorner(j_2) \Leftrightarrow v'_{j_1} \in \ulcorner \phi_u \lrcorner(v'_{j_2})$$

We shall in fact prove that $\ulcorner \phi_u \lrcorner(v'_{j_2}) \setminus \{\phi_u(i), i\} = \{v'_x \mid x \in \ulcorner \phi_{v'} \lrcorner(j_2)\}$, which clearly implies the proposition above.

We distinguish two cases, depending on the polarity of v'_{j_2}.

- Either $v'_{j_2} \equiv_2 i$. In this case, $i \notin \ulcorner \phi_u \lrcorner(v'_{j_2})$ since that would imply that $i - 1 \in \ulcorner \phi_u \lrcorner(v'_{j_2})$, which is impossible by Lemma 5. Thus $\ulcorner \phi_u \lrcorner(v'_{j_2}) \cap \{\phi_u(i), \ldots, i\} = \emptyset$, and we even have $\ulcorner \phi_u \lrcorner(v'_{j_2}) = \{v'_x \mid x \in \ulcorner \phi_{v'} \lrcorner(j_2)\}$.
- Or $v'_{j_2} \not\equiv_2 i$. In this case, a straightforward case analysis gives that, if $\ulcorner \phi_u \lrcorner(v'_{j_2}) \cap \{\phi_u(i), i\} \neq \emptyset$, then $\{\phi_u(i) - 1, i + 1\} \subseteq \ulcorner \phi_u \lrcorner(v'_{j_2})$. Thus $\ulcorner \phi_u \lrcorner(v'_{j_2}) \setminus \{\phi_u(i), i\} = \{v'_x \mid x \in \ulcorner \phi_{v'} \lrcorner(j_2)\}$.

Thus, $v' \in VSub(\phi_u)$, from which it is straightforward to check that $v = (u_{v'_i})_{I-(i-\phi_u(i)+1)}$ is a visible subsequence of ϕ.

It remains to check that $v \prec^* u$. This is justified by the fact that until $\phi_u(i)$, u and v coincide. Then,

- Either $i + 1 \in I$, then by construction, $v_{\phi_u(i)} = u_{i+1}$. But then:

$$\begin{aligned} \ulcorner \phi_u \lrcorner(i+1) &= \{i+1\} \cup \llcorner \phi_u \urcorner(i) \\ &= \{i+1, i\} \cup \ulcorner \phi_u \lrcorner(\phi_u(i)) \end{aligned}$$

  Thus $\phi_u(i) \in \ulcorner \phi_u \lrcorner(i+1)$. Since u is a visible subsequence of ϕ, we deduce that $u_{\phi_u(i)} \in \ulcorner \phi \lrcorner(u_{i+1})$, thus $u_{i+1} \prec u_{\phi_u(i)}$, thus $v \prec^* u$.
- Otherwise u_{i+1} is undefined. Then, v is only defined up to $\phi_u(i) - 1$ and is shorter than u, thus $v \prec^* u$. □

We immediately deduce the normalization theorem:

Theorem 3. *Let ϕ be an ultimately visible pointer structure on which $\prec$ is well-founded. Then ϕ is finite.*

Proof. Suppose ϕ is infinite. Then by Lemma 4, it has a minimal infinite visible subsequence m^ϕ. Then m^ϕ is such that, for any m^ϕ_i, there is j such that $\phi(m^\phi_j) = m^\phi_i$. Indeed, if this is not the case, Lemma 6 would build an infinite visible subsequence u of ϕ such that $u \prec^* m^\phi$. But then we could build an infinite sequence $x_0, x_1, \ldots$ of moves of ϕ such that, for all $i \in \mathbb{N}$, $x_i = \phi(x_{i+1})$. It is then straightforward to check that (if N is such that ϕ is visible after N) for all i such that $x_i > N$, $x_{i+2} \prec x_i$, which immediately conflicts with the well-foundedness of $\prec$. □

The formulation above is different from the one we announced in Section 2, hence let us recall and prove it.

Theorem 2 (revisited). *Let ϕ be an ultimately visible pointer structure with no infinite pointer chain. Then, if ϕ has forks of arbitrary size, it has a conscious fork of infinite size.*

Proof. If ϕ has finite depth and arbitrarily large forks, it is in particular infinite. Therefore, by Theorem 3, $\prec$ cannot be well-founded on ϕ. A counterexample to well-foundedness is an infinite sequence $x_0 \succ x_1 \succ x_2 \succ \ldots$. By definition of $\prec$, we then have $\ulcorner\phi_\lrcorner(x_0) \subset \ulcorner\phi_\lrcorner(x_1) \subset \ulcorner\phi_\lrcorner(x_2) \subset \ldots$ so we can take the limit:

$$X = \bigcup_{n\in\mathbb{N}} \ulcorner\phi_\lrcorner(x_n)$$

X has a tree structure, considering that an integer i has for sons all j such that $\phi(j) = i$. This tree cannot have an infinite branch (since ϕ has no infinite pointer chain) and is infinite, thus it has a node with infinite degree by König's lemma. By construction, each of the sons of this node with infinite degree sees all the prior ones, thus giving an infinite conscious fork. □

5. Consequences

This brief section lists some corollaries of our finiteness theorems specific to HO/N game semantics. As in several earlier parts of this paper, we assume familiarity with the vocabulary of HO game semantics.

Let us first define some subclasses of innocent strategies on arena games. (While the restriction to innocent strategies is convenient, it is not strictly necessary—but a general treatment would go beyond the scope of this paper.)

Definition 11. *Let A be an arena. An innocent strategy $\sigma : A$ is:*

- Total *iff, for all $sa \in \mathcal{L}_A$ such that $s \in \sigma$, there is b such that $sab \in \sigma$;*
- Finite *iff it has a finite view function;*
- Bounded *iff there is $N \in \mathbb{N}$ such that, for any play $s \in \sigma$, $|\ulcorner s \urcorner| \leq N$.*
- Noetherian *iff there is no strictly increasing infinite sequence of P-views $s_1 \sqsubset s_2 \sqsubset s_3 \sqsubset \ldots$ in σ.*

We also need some technical lemmas. The first is an obvious, quantitative strengthening of Theorem 1; the others provide an analysis of the interactions between σ and τ that give rise to the P-views of $\sigma;\tau$.

Lemma 7. *If $n, p \in \mathbb{N}$, there is some $T(n,p) \in \mathbb{N}$ such that, for all $\phi \in n \star p$, $|\phi| \leq T(n,p)$ where $|\phi|$ is the size of ϕ,* i.e. *the cardinal of $\mathcal{D}_\phi$.*

Proof. The set $n \star p$ has a tree structure, given by immediate prefix, which is obviously finitely branching. It has no infinite branch since that would produce an infinite pointer structure in $n \star p$, which is forbidden by Theorem 1. So, by König's lemma, it is finite and we define $T(n,p)$ to be its depth. □

Let $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$ be innocent strategies. A *pre-view* of $\sigma;\tau$ is an interaction $u \in \sigma||\tau$ with only one initial move on C and such that the external Opponent always points to the previous move, *i.e.* $u_{\restriction A,C}$ is a P-view. A pre-view is always a (potentially infinite) legal play for $(A \Rightarrow B) \Rightarrow C$ and, as such, induces a pointer structure ϕ_u.

The key property of a pre-view is the *switching condition*: only Player can switch between B and C whereas only Opponent can switch between A and B. As a consequence, if u ends with a move in B or C, its P-view only visits B and C and is a P-view of $B \Rightarrow C$; whereas, if it ends with a move in A, $\ulcorner u \urcorner_{\restriction A,B}$ is an O-view of $A \Rightarrow B$ where each move in A points to its preceding move. Dually, $\llcorner u \lrcorner_{\restriction A,B}$ is a P-view of $A \Rightarrow B$; whereas $\llcorner u \lrcorner_{\restriction B,C}$ is an O-view of $B \Rightarrow C$ where each move in C points to its preceding move.

Lemma 8. *If the sizes of P-views of σ and τ are bounded respectively by N_σ and N_τ then $\phi_u \in (N_\sigma + N_\tau) \star \max(N_\sigma + 1, N_\tau)$.*

Proof. The general form of a P-view of ϕ_u is $s \cdot t$ where s is a P-view of τ and t is an O-view of A which is also a P-view. This means that t is a subsequence of a P-view of σ. Hence lengths of P-views of ϕ_u are bounded by $N_\sigma + N_\tau$.

An O-view of ϕ_u is either an O-view of C that is also a sub-P-view of τ and is therefore bounded by N_τ; or is of the form $c \cdot s$, where c is an initial move of C and s is a P-view of σ and is thus bounded by $N_\sigma + 1$. So lengths of O-views of ϕ_u are bounded by $\max(N_\sigma + 1, N_\tau)$. □

Lemma 9. *If σ and τ are noetherian then the relation $\prec$ induced by ϕ_u is well-founded.*

Proof. An infinite increasing sequence in ϕ_u is an infinite increasing sequence of either P- or O-views. In the case of P-views, the same reasoning as in Lemma 8 implies that this would induce either an infinite increasing sequence of P-views of τ or an infinite increasing sequence (t_i) of sub-P-views of σ.

The first case is immediately ruled out because τ is noetherian. In the second case, the sequence (t_i) would determine a strictly increasing sequence (s_i) of prefixes of $u_{\restriction A,B}$ where each $s_i \in \sigma$ (and so $\ulcorner s_i \urcorner \in \sigma$) and each s_j sees all s_is with $i < j$. So $(\ulcorner s_i \urcorner)$ would be strictly increasing which is ruled out because σ is noetherian.

The argument for O-views is completely symmetric. □

Lemma 10. *If σ and τ are bounded (resp. finite, noetherian) then so is $\sigma;\tau$.*

Proof. If σ and τ are bounded, let P be the set of *pre-views* of $\sigma;\tau$. By Lemma 8, each $u \in P$ gives rise to $\phi_u \in (N_\sigma + N_\tau) \star \max(N_\sigma + 1, N_\tau)$ and, by Lemma 7, its length is bounded by $T(N_\sigma + N_\tau, \max(N_\sigma + 1, N_\tau))$. Since this bound is uniform for all $u \in P$ and restricting to $A \Rightarrow C$ can only reduce their size, the size of all P-views of $\sigma;\tau$ is bounded by $T(N_\sigma + N_\tau, \max(N_\sigma + 1, N_\tau))$.

If σ and τ are finite, they are in particular bounded and so $\sigma;\tau$ is too. Moreover, there must be finite restrictions A', B' and C' of the arenas A, B and C such that we still have $\sigma : A' \Rightarrow B'$ and $\tau : B' \Rightarrow C'$. So $\sigma;\tau$ is a bounded strategy on the finite arena $A' \Rightarrow C'$ and, by König's lemma, must be finite.

Finally, suppose σ and τ are noetherian and that there is an infinite increasing sequence of P-views in $\sigma;\tau$. By determinism of σ and τ, this would induce an infinite increasing sequence of pre-views $u_1 \sqsubset u_2 \sqsubset u_3 \sqsubset \cdots \in \sigma||\tau$ whose limit is an infinite pre-view u. By Lemma 9, the relation $\prec$ induced by ϕ_u would be well-founded—which is impossible by Theorem 3. □

Proposition 5. *If σ and τ are both bounded (resp. noetherian) and total then $\sigma;\tau$ is total.*

Proof. For a contradiction, let us suppose that $\sigma;\tau$ is not total. So we can find an infinite interaction in $\sigma||\tau$ of the form uu_∞ such that $u_{\restriction A,C}$ is finite and $u_{\restriction A,C} = uu_{\infty\restriction A,C}$, *i.e.* u_∞ is an infinite tail in B. Since σ and τ are innocent, the infinite pre-view $u'_\infty := \ulcorner u \urcorner u_\infty \in \sigma||\tau$. (By definition, $\ulcorner u \urcorner$ is the *view* [13] of u that satisfies $\ulcorner u \urcorner_{\restriction A,C} = \ulcorner u_{\restriction A,C} \urcorner$.)

In the case where σ and τ are bounded, Lemma 8 guarantees us a finite bound for $\phi_{u'_\infty}$; and in the case where they are noetherian, Lemma 9 permits us to apply Theorem 3 to obtain a finite bound for $\phi_{u'_\infty}$. In either case, we have our desired contradiction. □

We have shown that bounded or noetherian total *innocent* strategies are preserved by composition. The dependency on innocence can be weakened, in the case of bounded strategies, to just the visibility condition (without even the need for determinism) but this does require considerable further technical development which goes beyond the scope of this paper. The details will appear in the first author's forthcoming PhD thesis.

On the other hand, the case of noetherian strategies is rather more subtle. It seems that the class of noetherian finitely nondeterministic (visible) strategies is closed under composition—despite the fact that neither the noetherian nor the finite nondeterminism constraint is preserved in isolation! The argument of Lemma 10 could then, in principle, be adapted to this more general setting by considering the *tree* of pre-views which share the same projection on $A \Rightarrow C$ and using an additional application of König's lemma to build the needed infinite pre-view, cf. Proposition 4.4.2 of [13]. The case of countable nondeterminism, however, seems resistant to this approach: the strategy `random` : `nat` that *may* return any, but *must* return some, integer, when composed with `repeat`, yields a fundamentally non-noetherian strategy.

6. Conclusion

We have presented a thorough-going analysis of totality in arena games, in particular proving the folklore theorem that total finite innocent strategies form a category. However, many natural situations involve non-finite strategies. For example, the finiteness hypothesis fails as soon as one try to models the integers—whatever the arena chosen, flat or recursive—and, in the case of the flat arena, this restriction is totally arbitrary since our theorems show that the width of arenas does not in any way influence the compositionality of totality. Thus finiteness of strategies can be relaxed to the existence of a bound on the length of their views, or even to the absence of infinite views.

The next step is to drop innocence. This may seem less natural, since the meaning of the "no infinite view" hypothesis for non-innocent strategies is not so clear, but we do have a natural example: the naive interpretation in HO/N arena games of nondeterministic terms provides such strategies. Moreover, this is not the only interesting case: the "clock" operator employed by Krivine [21] to realize the axiom of dependent choice probably also behaves as such.

Finally, the decomposition of having finite views into finite depth and finite memory presented in Section 2 opens up the possibility of having compositionality of totality for a class of strategies with infinite views. Since no interesting λ-term generates an infinite conscious fork (as this would require some variable to occur infinitely often in its Böhm tree), we could use an external argument (such as a typical winning condition) to guarantee the finite depth of an interaction. This could be technically lighter than directly forcing finiteness of the interaction and would be sufficient as our results would then imply its finiteness. Such strategies without infinite conscious fork, but with infinite views, appear as soon as one wants to model polymorphism [12] or languages with inductive/coinductive definitions [4].

Some of the arguments presented in this paper, particularly the last (strong) finiteness theorem, could easily be carried out in the usual vocabulary of arena games. However, we believe that the abstract language of pointer structures is more convenient since it applies smoothly to a large number of settings where pointers arise naturally, including of course the model of Hyland and Ong for PCF [20] and the large number of games models based on it, but also traces of abstract machines like the PAM and others.

The original goal which motivated this work was the construction of a locally cartesian closed category of games and total innocent strategies in order to give game semantics of dependent type theories [27]. This work can now be continued. Another interesting idea would be to investigate how our results interact with the new construction of the category of arena games and innocent strategies in [15], where pointer structures play a central role.

Acknowledgements. We would like to thank our colleagues Pierre-Louis Curien, Olivier Laurent and Laurent Régnier for the numerous stimulating discussions we had during the development of this work; and the anonymous referees whose comments have significantly improved the original draft.

References

[1] S. Abramsky. Semantics of interaction: an introduction to game semantics. *Semantics and Logics of Computation*, pages 1–31, 1996.

[2] S. Abramsky, K. Honda, and G. McCusker. A fully abstract game semantics for general references. In *Proceedings, Thirteenth Annual IEEE Symposium on Logic in Computer Science*, 1998.

[3] S. Abramsky and G. McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions. In P. W. O'Hearn and R. D. Tennent, editors, *Algol-like languages*. Birkhaüser, 1997.

[4] D. Baelde and D. Miller. Least and greatest fixed points in linear logic. *Lecture Notes in Computer Science*, 4790:92, 2007.

[5] T. Coquand. A semantics of evidence for classical arithmetic. *Journal of Symbolic Logic*, 60(1):325–337, 1995.

[6] P.-L. Curien. Abstract Böhm trees. *Mathematical Structures in Computer Science*, 8(06):559–591, 1998.

[7] V. Danos and R. Harmer. Probabilistic game semantics (extended abstract). In *Proceedings, 15th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 2000.

[8] V. Danos, H. Herbelin, and L. Regnier. Game semantics and abstract machines. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science*, pages 394–405, 1996.

[9] V. Danos and L. Regnier. How abstract machines implement head linear reduction. *Submitted to Higher Order and Symbolic Computation*, 2004.

[10] J. de Lataillade. Second-order type isomorphisms through game semantics. *Annals of Pure and Applied Logic*, 151:115–150, 2008.

[11] C. Faggian and J.M.E. Hyland. Designs, disputes and strategies. *Lecture notes in computer science*, pages 442–457, 2002.

[12] J.-Y. Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45(2):159–192, 1986.

[13] R. Harmer. *Games and full abstraction for nondeterministic languages*. PhD thesis, University of London, 1999.

[14] R. Harmer. Innocent game semantics. Lecture notes, 2004–2007.

[15] R. Harmer, J.M.E. Hyland, and P.-A. Mellies. Categorical Combinatorics for Innocent Strategies. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*, pages 379–388. IEEE Computer Society, Washington DC, USA, 2007.

[16] R. Harmer and O. Laurent. The anatomy of innocence revisited. In *Proceedings, 26th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer Verlag, 2006.

[17] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[18] D. Hughes. Games and definability for System F. *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 76–86, 1997.

[19] J.M.E. Hyland. Game semantics. *Semantics and Logics of Computation*, 1997.

[20] J.M.E. Hyland and C.H.L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation*, 163:285–408, 2000.

[21] J.-L. Krivine. Dependent choice, 'quote' and the clock. *Theoretical Computer Science*, 308(1-3):259–276, 2003.

[22] J. Laird. Full abstraction for functional languages with control. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science*, pages 58–67. IEEE Computer Society Press, 1997.

[23] O. Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1–3):79–123, December 2004.

[24] G. McCusker. Games and definability for FPC. *Bulletin of Symbolic Logic*, 3(3):347–362, September 1997.

[25] R. Milner. *Communication and concurrency*. Prentice Hall, 1989.

[26] H. Nickau. Hereditarily sequential functionals. *Lecture Notes in Computer Science*, 813:253–264, 1994.

[27] R.A.G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(33-48):3–6, 1984.